Robustness and Accuracy: Are We Trying to Have Our Cake and Eat It Too?

Abstract

The phenomenon of training robust deep neural networks coinciding with reduced accuracy has extensively manifested itself empirically. We look at theoretical explanations for why this observed tradeoff may be inherent and examine a state-of-the-art defense method, designed trade adversarial robustness off against accuracy. We also consider the perspective that models which are both robust and accurate can be obtained, and evaluate if batch normalization can be of benefit in this pursuit.

1 Introduction

Deep neural networks have become increasingly popular and have been deployed in many machine learning tasks in the domains of images, text and speech, with remarkable success. These include safety-critical applications such as road-sign recognition in autonomous vehicles ([3]). Simultaneously many works ([17], [5], [7]) have shown that DNNs are vulnerable to adversarial examples. To deal with this a number of countermeasures have been considered ([14], [12]). However, when implemented these have reduced the accuracy on standard examples ([16]). This leads us to question if the apparent trade-off is inevitable.

2 Notations

We consider an input $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^d$, a binary or multi-class classification problem with labels $y \in \mathcal{Y} := \{-1, +1\}$ or $\mathcal{Y} := \{1, \ldots, C\}$, sampled from an unknown distribution $(\mathbf{X}, Y) \sim \mathcal{D}$. The labels partion $\mathcal{X}$ into disjoint classes $\mathcal{X}^-, \mathcal{X}^+$ or $\mathcal{X}^{(1)}, \ldots, \mathcal{X}^{(C)}$. For a classifier $g : \mathcal{X} \to \mathcal{Y}$ the natural error is defined as $\mathcal{R}_{nat}(g) := \mathbb{E}_{(\mathbf{X},Y)\sim\mathcal{D}}\mathbf{1}\{g(\mathbf{X}) \neq Y\}$ and the robust error under the threat of an ϵ-bounded perturbation as $\mathcal{R}_{rob}(g) := \mathbb{E}_{(\mathbf{X},Y)\sim\mathcal{D}}\mathbf{1}\{\exists \mathbf{X}' \in \mathbb{B}(\mathbf{X}, \epsilon)\, s.t.\, g(\mathbf{X}') \neq Y\}$. For all classifiers g, $\mathcal{R}_{rob}(g) \geq \mathcal{R}_{nat}(g)$, with equality if $\epsilon = 0$. In the context of binary classification, given a score function $f : \mathcal{X} \to \mathbb{R}$ we will extend our notation writing $\mathcal{R}_{nat}(f) := \mathbb{E}_{(\mathbf{X},Y)\sim\mathcal{D}}\mathbf{1}\{\text{sign}(f(\mathbf{X})) \neq Y\}$ and $\mathcal{R}_{rob}(f) := \mathbb{E}_{(\mathbf{X},Y)\sim\mathcal{D}}\mathbf{1}\{\exists \mathbf{X}' \in \mathbb{B}(\mathbf{X}, \epsilon)\, s.t.\, \text{sign}(f(\mathbf{X}')) \neq Y\}$ for associated natural and robust errors. The natural and robust accuracies are defined as $\mathcal{A}_{nat} = 1 - \mathcal{R}_{nat}$ and $\mathcal{A}_{rob} = 1 - \mathcal{R}_{rob}$ respectively. $\|\cdot\|$ represents a generic norm, we specify when required e.g. $\|\cdot\|_2$. We define $\|\mathbf{x} - \mathcal{X}^{(i)}\| = \min_{\mathbf{x}' \in \mathcal{X}^{(i)}} \|\mathbf{x} - \mathbf{x}'\|$ and $\|\mathcal{X}^{(i)} - \mathcal{X}^{(j)}\| = \min_{\mathbf{x} \in \mathcal{X}^{(i)}, \mathbf{x}' \in \mathcal{X}^{(j)}} \|\mathbf{x} - \mathbf{x}'\|$. We limit ourselves to classification tasks with l_p-bounded adversaries.

3 Robustness May Be at Odds with Accuracy

We commence by examining [19], in this paper the following input-label distribution is constructed:

$$y \overset{u.a.r.}{\sim} \{-1, +1\}, \qquad x_1 = \begin{cases} +y & w.p. \quad p \\ -y & w.p. \quad 1-p \end{cases}, \qquad x_2, \ldots x_{d+1} \overset{i.i.d.}{\sim} N(\eta y, 1),$$

where $0.5 \leq p < 1$ and η is chosen to be large enough e.g. $\eta = O(\frac{1}{\sqrt{d}})$ will suffice. For this distribution, a linear classifier $g_{avg}(\mathbf{x}) := \text{sign}(\omega_{unif}^\top \mathbf{x})$, where $\omega_{unif} := [0, \frac{1}{d}, \ldots, \frac{1}{d}]$, achieves

32nd Conference on Neural Information Processing Systems (NeurIPS 2018), Montréal, Canada.

natural accuracy arbitrarily close to 100%, for sufficiently large d. This classifier does not consider x_1 and instead relies on the weakly correlated features $x_2, \ldots, x_{d+1}$, obtaining high accuracy through implicitly pooling and treating $x_2, \ldots, x_{d+1}$, as a single "meta-feature". Now consider a l_∞-bounded perturbation with $\epsilon = 2\eta$, which shifts each $x_2, \ldots, x_{d+1}$ towards $-y$. The resulting features $x'_2, \ldots, x'_{d+1}$ of the perturbed input x' are sampled i.i.d. from the anti-correlated distribution $-N(\eta y, 1)$. Consequently, the robust accuracy of g_{avg} will be arbitrarily close to 0%. We can draw a distinction between robust features x_1 and non-robust features $x_2, \ldots, x_{d+1}$ ande note that any classifier which achieves a natural accuracy $> p$ has to at least in part rely on non-robust features, this allows the authors to establish a generalisation of the trade-off between natural and robust accuracy demonstrated by g_{avg} to any classifier:

Theorem 3.1 *Any classifier that attains at least $1 - \delta$ standard accuracy on $\mathcal{D}$ has robust accuracy at most $\frac{p}{1-p}\delta$ against an l_∞-bounded adversary with $\epsilon \geq 2\eta$.*

Since $p < 1$, as $\delta \to 0$ and natural accuracy approaches 100%, adversarial accuracy falls to 0% for any classifier just as seen with g_{avg}. Numerically, if $p = 0.95$, all classifiers with natural accuracy $\geq 96\%$, $\geq 99\%$ have robust accuracy $\geq 76\%$ and $\geq 19\%$ respectively. The perturbation required to achieve this is small relative to the size of the input ($\|\mathbf{x}\|_\infty \geq 1$) with $\epsilon = O(\eta)$ and $\eta = O(\frac{1}{\sqrt{d}})$. Thus we have a concrete example for which the robustness-accuracy tradeoff has been proven. The result is particularly notable as it holds even in the regime of infinite data. This combined with the empirical results in the appendix of this paper and the extremely extensive empirical results in [16], which after inspecting 18 deep image classification models against several attacks concluded "more accurate network models appear to be less robust in terms of the required adversarial attack strength defined in l_p ball", may lead to the viewpoint that the afromentioned tradeoff is inherent. The hypothesis that highly accurate classifiers learn non-robust features can also explain the phenomenon of transferability of adversarial attacks ([17], [16]). It also demonstrates the necessity of adversarial training algorithms which learn robust features when robustness is the goal.

4 Theoretically Principled Trade-off between Robustness and Accuracy

Taking the above perspective, [23] introduces a new method for adversarial training which the authors name **TRADES** (TRadeoff-inspired Adversarial DEfense via Surrogate-loss minimization). Instead of 0-1 loss which is non-differentiable, making any optimisation challenging, a surrogate loss function ϕ is considered. To derive the following bounds a weak assumption on ϕ is made - it is classification-calibrated ([1]). Table 1 lists some common surrogate loss functions that satisfy this assumption and states their associated ψ-transforms. For a score function $f : \mathcal{X} \to \mathbb{R}$, let $\mathcal{R}^*_{nat} = \min_f \mathcal{R}_{nat}(f)$, $\mathcal{R}_\phi(f) = \mathbb{E}(\phi(f(\mathbf{X}), Y))$ and $\mathcal{R}^*_\phi = \min_f \mathcal{R}_\phi(f)$, then:

Theorem 4.1 *For any non-negative classification-calibrated loss function ψ s.t. $\psi(0) \geq 1$, any measurable $f : \mathcal{X} \to \mathbb{R}$, any probability distribution on $\mathcal{X} \times \{-1, +1\}$, and any $\lambda > 0$, we have*

$$\mathcal{R}_{rob}(f) - \mathcal{R}^*_{nat} \leq \psi^{-1}(\mathcal{R}_\phi(f) - \mathcal{R}^*_\phi) + \mathbb{E}(\max_{\mathbf{X}' \in \mathbb{B}(\mathbf{X}, \epsilon)} \phi(f(\boldsymbol{X})f(\boldsymbol{X}')/\lambda))$$

Theorem 4.2 *Suppose that $|\mathcal{X}| \geq 2$. For any non-negative classification-calibrated loss function ψ s.t. $\psi(x) \to 0$ as $x \to \infty$, any $\xi > 0$, and any $\theta \in [0, 1]$, there exists a probability distribution on $\mathcal{X} \times \{-1, +1\}$, a function $f : \mathbb{R}^d \to \mathbb{R}$, and a regularization parameter $\lambda > 0$ such that $\mathcal{R}_{rob}(f) - \mathcal{R}^*_{nat} = \theta$ and*

$$\psi\Big(\theta - \mathbb{E}(\max_{\mathbf{X}' \in \mathbb{B}(\mathbf{X}, \epsilon)} \phi(f(\boldsymbol{X})f(\boldsymbol{X}')/\lambda))\Big) \leq \mathcal{R}_\phi(f) - \mathcal{R}^*_\phi \leq \psi\Big(\theta - \mathbb{E}(\max_{\mathbf{X}' \in \mathbb{B}(\mathbf{X}, \epsilon)} \phi(f(\boldsymbol{X})f(\boldsymbol{X}')/\lambda))\Big) + \xi$$

Thus given $\psi(x) \to 0$ as $x \to \infty$ (which holds for all ψ in Table 1) Theorem 4.2 demonstrates that the bound in Theorem 4.1 is tight. The optimality of Theorem 4.1 is also shown empirically by fitting a classifier on the MINST dataset with $\mathcal{R}^*_{nat} = 0$ and the expectation estimated over the test data. Relying on λ to approximately reflect the effect of ψ^{-1}, when $\psi(\theta) \neq \theta$, Theorem 4.1 suggests the following method of optimization (as $\mathcal{R}^*_{nat}$ is independent of f):

$$\min_f \mathbb{E}\Big\{ \underbrace{\phi(f(\mathbf{X})Y)}_{\text{accuracy}} + \underbrace{\max_{\mathbf{X}' \in \mathbb{B}(\mathbf{X}, \epsilon)} \phi(f(\mathbf{X})f(\mathbf{X}')/\lambda)}_{\text{robustness}} \Big\}$$

Table 1: Classification-calibrated surrogate loss ϕ and its associated ψ-transform. Here $\psi_{log}(\theta) = \frac{1}{2}(1-\theta)log_2(1-\theta) + \frac{1}{2}(1+\theta)log_2(1+\theta)$.

Loss	$\phi(\alpha)$	$\psi(\theta)$
Hinge	$max\{1-\alpha, 0\}$	θ
Sigmoid	$1 - tanh(\alpha)$	θ
Exponential	$exp(-\alpha)$	$1 - \sqrt{1-\theta^2}$
Logistic	$log_2(1 + exp(-\alpha))$	$\psi_{log}(\theta)$

The first term reduces natural error encouraging accuracy. Minimising the second term moves the boundary of the classifier away from the sample instances, thus they become less susceptible to small norm perturbations, hence encouraging robustness. The parameter λ balances these goals, with higher values prioritising natural accuracy and lower values prioritising robust accuracy. In order to extend to more general defenses, the authors heuristically allow for multi-class classification via replacing ψ with a multi-class calibrated loss $\mathcal{L}$ and approximately solving the minimax problem via alternating gradient descent. A widely utilized example of a multi-class calibrated loss is cross-entropy loss ([15]). This gives:

$$\min_f \mathbb{E}\Big\{\mathcal{L}(f(\mathbf{X}), \mathbf{Y}) + \max_{\mathbf{X}' \in \mathbb{B}(\mathbf{X}, \epsilon)} \mathcal{L}(f(\mathbf{X}), f(\mathbf{X}')/\lambda)\Big\}$$

where $f(\mathbf{X})$ is the output vector (with softmax operator in the final layer) and $\mathbf{Y}$ is the label-indicator vector. The pseudocode, that aims to minimize the empirical form of the above, is displayed in Algorithm 1. Projected gradient descent in Step 7 should be highlighted, as it is essential we initialize $\mathbf{x}'_i$ by adding a small, random perturbation around $\mathbf{x}_i$ for inner maximization. This is done as $\mathbf{x}_i$ is the global minimum with zero gradient of $g(\mathbf{x}') = \mathcal{L}(f(\mathbf{x}_i), f(\mathbf{x}'))$. The resulting new form of surrogate loss for adversarial training performed exceptionally on MNIST and CIFAR10 datasets against a variety of white-box and black-box threat models, won the NeurIPS 2018 Adversarial Vision Challenge with over 2,000 submissions and is still refered to as "state-of-the-art" ([8]).

Algorithm 1 Adversarial training by TRADES

1: **Input:** Step sizes η_1 and η_2, batch size m, number of iterations K in inner optimization, network architecture parametrized by θ
2: **Output:** Robust network f_θ
3: Randomly initialize network f_θ, or initialize network with pre-trained configuration
4: **repeat**
5: Read mini-batch $B = \{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$ from training set
6: **for** $i = 1, \ldots, m$ (in parallel) **do**
7: $\mathbf{x}'_i \leftarrow \mathbf{x}_i + 0.001 \cdot N(0, I_d)$
8: **for** $k = 1, \ldots, K$ **do**
9: $\mathbf{x}'_i \leftarrow \Pi_{\mathbb{B}(\mathbf{x}_i, \epsilon)}(\eta_1 \text{sign}(\nabla_{\mathbf{x}'_i} \mathcal{L}(f_\theta(\mathbf{x}_i), f_\theta(\mathbf{x}'_i))) + \mathbf{x}'_i)$, here Π is the projection operator
10: **end for**
11: **end for**
12: $\theta \leftarrow \theta - \eta_2 \sum_{i=1}^m \nabla_\theta [\mathcal{L}(f_\theta(\mathbf{x}_i, \mathbf{y}_i) + \mathcal{L}(f_\theta(\mathbf{x}_i), f_\theta(\mathbf{x}'_i))/\lambda]/m$
13: **until** training converged

5 A Closer Look at Accuracy vs. Robustness

The authors of [21], take a different viewpoint. For a data distribution they introduce:

Definition 5.1 (r-**separation**) *A data distribution over* $\bigcup_{i \in \{1, \ldots, C\}} \mathcal{X}^{(i)}$ *is* r*-separated if* $\|\mathcal{X}^{(i)} - \mathcal{X}^{(j)}\| \geq 2r$ *for all* $i \neq j$.

Given r-separation, it is theoretically established that a classifier which achieves perfect robust (with radius r) and natural accuracy exists. In the binary and multi-class classification case consider the functions $f_{bin}(\mathbf{x}) = \frac{\|\mathbf{x} - \mathcal{X}^-\| - \|\mathbf{x} - \mathcal{X}^+\|}{2r}$ and $f_{mc}(\mathbf{x}) = \frac{1}{r} \cdot (\|\mathbf{x} - \mathcal{X}^{(1)}\|, \ldots, \|\mathbf{x} - \mathcal{X}^{(C)}\|)$ respectively. By directly showing that these functions satisfy the requirements, the following theorems are proved:

Table 2: Separation of real data. Perturbation radii ϵ are those used in [20], [18] and [13] respectively.

	Perturbation radii ϵ	Minimum Train-Train separation	Minimum Test-Train separation
Fashion-MNIST	0.1	0.318	0.322
Chest X-Ray Images	0.1	0.482	0.463
GTSRB	0.1	0.035	0.055

Theorem 5.1 *Suppose $\mathcal{X} = \mathcal{X}^- \cup \mathcal{X}^+$ is r-separated. Then, there exists $f : \mathcal{X} \to \mathbb{R}$ s.t.*
(a) f is locally Lipschitz with constant $\frac{1}{r}$ on a ball of radius r around all $\mathbf{x} \in \mathcal{X}$, and
(b) the classifier $g = sign(f)$ has $\mathcal{A}_{nat}(g) = 1$ and $\mathcal{A}_{rob}(g) = 1$ with $\epsilon = r$.

Theorem 5.2 *Suppose $\mathcal{X} = \bigcup_{i\in\{1,\dots,C\}} \mathcal{X}^{(i)}$ is r-separated. Then, there exists $f : \mathcal{X} \to \mathbb{R}^C$ s.t.*
(a) f is locally Lipschitz with constant $\frac{1}{r}$ on a ball of radius r around all $\mathbf{x} \in \mathcal{X}$, and
(b) the classifier $g = argmin_{i\in\{1,\dots,C\}}(f)_i$ has $\mathcal{A}_{nat}(g) = 1$ and $\mathcal{A}_{rob}(g) = 1$ with $\epsilon = r$.

The paper argues that real image datasets are r-separated, with $\epsilon \ll r$, where ϵ is the typical size of adversarial perturbation. Specifically, MNIST, CIFAR-10, SVHN and Restricted ImageNet datasets are considered with respect to l_∞-distance. Let the Train-Train Separation be the minimum distance between two examples in the training set with different class labels, and the Test-Train Separation be the minimum distance between an example in the test set and an example in the training set with a different class label. Upon removing four pairs of duplicate examples with multiple labels and three examples which when examined are visibly highly noisy, both the Train-Train and Test-Train separations are higher than 2ϵ for all considered datasets. A limitation of the results is that they are empirical, whilst the definition of r-separation refers to the underlying distribution of the data, which is unknown. However considering the size of the datasets, it is reasonable to assume that the data supports are representative of the support of the distribution. To exemplify the theoretical existence, for the MNIST, CIFAR-10 and SVHN datasets, proof-of-concept neural networks are also constructed, which achieve $\mathcal{A}_{nat}$ and $\mathcal{A}_{rob} \geq 0.999$.

It can also be argued that the choice of datasets lacks scope, with MINST and SVHN, as well as CIFAR-10 and Restricted ImageNet (containing only nine animal classes), being very similar. Consequently, we choose to examine three further datasets: Fashion-MNIST, Chest X-Ray Images (Pneumonia) and GTSRB(German Traffic Sign Recognition Benchmark). The former was chosen due to its high usage in academic papers and the latter two due to their relevance to practical applications of deep learning. Our findings are presented in Table 2. We verify that Fashion-MNIST and Chest X-Ray Images datasets are r-separated with sufficiently large r. This is not the case for GTSRB, at least relative to ϵ values used in [13]. This is also not solved by removing a few examples as 7221 images in the test set and 2236 in images in the train set (out of 39209 and 12630) have a neighbor in the train set with a different label less than 2ϵ away. Nonetheless a choice of $\epsilon = 0.005$ which is used with Restricted ImageNet would satisfy $r < 2\epsilon$ (empirically). Finally, we note that the distribution constructed in [19] does not contradict Theorem 5.1, $p < 1$ and the Guassians have infinite support.

The paper proposes two possible explanations for why the trade-off has been observed in experimental results. One is that the current training methods fail to impose the local Lipschitzness properly and another may be that they fail to generalise properly. Their experimental results largely support these ideas with more robust methods imposing higher degrees of local Lipschitzness, but also have larger natural and adversarial generalization gap. The authors observe that dropout reduces the gap.

6 Our Experimental Results

Taking the framework provided by [21] we consider another model adjustment - batch normalisation (BN) ([9]). In [9] BN was shown to improve natural accuracy, [4] and [2] however demonstrated that it produces adversarial vulnerable models. [2] further argued that BN produces models to rely more on non-robust features to explain their findings. This work is novel (to our best knowledge), the impact of BN on natural robustness, adversarial robustness and local Lipschitzness of classifiers produced by TRADES, has not been considered prior. We use the same setup as the experiment on

Table 3: Impact of implementing BN.

	Architecture	Train acc.	Test acc.	Gap	Adv. train acc.	Adv. test acc.	Adv. gap	Test lipschitz
Natural	CNN1	100.00	99.16	0.84	61.37	61.25	0.12	50.39
Natural	CNN1_BN1	100.00	99.27	0.73	55.42	55.65	-0.23	73.82
Natural	CNN1_BN2	100.00	99.26	0.74	32.71	34.11	-1.4	91.24
TRADES($\frac{1}{\lambda} = 0.5$)	CNN1	99.92	99.28	0.64	98.5	96.23	2.27	10.87
TRADES($\frac{1}{\lambda} = 0.5$)	CNN1_BN1	99.97	99.3	0.67	99.08	96.34	2.74	10.17
TRADES($\frac{1}{\lambda} = 0.5$)	CNN1_BN2	99.97	99.25	0.72	99.24	96.54	2.7	9.63
TRADES($\frac{1}{\lambda} = 1$)	CNN1	99.81	99.25	0.56	98.76	96.71	2.05	9.24
TRADES($\frac{1}{\lambda} = 1$)	CNN1_BN1	99.84	99.22	0.62	99.00	96.87	2.13	7.93
TRADES($\frac{1}{\lambda} = 1$)	CNN1_BN2	99.87	99.24	0.63	99.06	96.79	2.27	8.12
TRADES($\frac{1}{\lambda} = 2$)	CNN1	99.51	99.14	0.37	98.59	96.78	1.81	7.86
TRADES($\frac{1}{\lambda} = 2$)	CNN1_BN1	99.52	99.13	0.39	98.68	96.95	1.73	6.68
TRADES($\frac{1}{\lambda} = 2$)	CNN1_BN2	99.48	99.16	0.32	98.45	96.98	1.47	6.23
TRADES($\frac{1}{\lambda} = 3$)	CNN1	99.22	98.97	0.25	97.91	96.45	1.46	7.66
TRADES($\frac{1}{\lambda} = 3$)	CNN1_BN1	99.14	98.95	0.19	97.81	96.67	1.14	5.28
TRADES($\frac{1}{\lambda} = 3$)	CNN1_BN2	99.04	98.84	0.2	97.66	96.73	0.93	4.57

MINST in [21] (network structure = CNN1, optimizer = SGD with momentum 0.9, batch size = 64, adversarial attack = PGD [10] with $\epsilon = 0.1$, running 160 epochs on the training dataset, decaying the learning rate by a factor 0.1 in the 40th, 80th 120th and 140th epochs, starting at 0.0001). CNN1 has two convolutional layers followed by two fully connected layers, with dropout layers preceding each of the fully connected layers. We consider adding BN after the first convolutional layer (CNN1_BN1) and after the first and second convolutional layer (CNN1_BN2). We do not consider adding BN after further layers as having dropout proceed BN has been shown to lead to adverse results ([11]). In [21] TRADES with $1/\lambda = 1, 3$ were shown to perform well, thus we consider $1/\lambda = 0.5, 1, 2$ and 3. We empirically measure the local Lipschitzness as: $\frac{1}{n}\sum_{i=1}^{n} \max_{x_i' \in \mathbb{B}_\infty(x_i,\epsilon)} \frac{\|f(x_i)-f(x_i')\|_1}{\|x_i-x_i'\|_\infty}$. The results are presented in Table 3. For natural learning they agree with previous findings, the implementation of BN improved natural accuracy and reduced robust accuracy. For TRADES, however robust accuracy tended to increase with each implementation of BN. In both the impact on robust accuracy agrees with the hypothesis and empirical results in [21], that robustness correlates positively with local Lipschitzness. Considering [2] we also can put forward the idea that by using an appropriate adversarial training method, the drawback of BN causing models to favour learning non-robust features may be alleviated.

7 Conclusion

Unfortunately, the question we asked ourselves in the introduction remains an open one. Whilst no classifier that is robust and accurate exists for the distribution in [19], it can be argued that this is not representative of the tasks faced by DNNs. After all the human eye has been widely assumed to be both accurate and impenetrable to small norm perturbations in typical settings, this can however be disputed as even for image classification DNNs can outperform humans ([6]). It also is hard to make the case that DNNs are not expressive enough to achieve this as they can fit not just adversarial but completely randomly labeled data perfectly [22]. At the same time obtaining a concrete result is challenging as it is likely to require some assumptions on the underlying distribution of the data, which in most real-world settings is unknown. Furthermore, even if we demonstrate that an accurate and classifier exists, we are left with the challenge of establishing a framework for finding it (although dropout and BN may be steps in the right direction). Notably, the classifiers used to prove results in [21] cannot be learned as they require knowledge of class supports.

References

[1] Peter L Bartlett, Michael I Jordan, and Jon D McAuliffe. "Convexity, classification, and risk bounds". In: *Journal of the American Statistical Association* 101.473 (2006), pp. 138–156.

[2] Philipp Benz, Chaoning Zhang, and In So Kweon. "Batch Normalization Increases Adversarial Vulnerability and Decreases Adversarial Transferability: A Non-Robust Feature Perspective". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 7818–7827.

[3] Dan CireAan et al. "Multi-column deep neural network for traffic sign classification". In: *Neural networks* 32 (2012), pp. 333–338.

[4] Angus Galloway et al. "Batch normalization is a cause of adversarial vulnerability". In: *arXiv preprint arXiv:1905.02161* (2019).

[5] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. "Explaining and harnessing adversarial examples". In: *arXiv preprint arXiv:1412.6572* (2014).

[6] Kaiming He et al. "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification". In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 1026–1034.

[7] Tianxing He and James Glass. "Detecting egregious responses in neural sequence-to-sequence models". In: *arXiv preprint arXiv:1809.04113* (2018).

[8] Yifei Huang et al. "Adversarial Robustness of Stabilized NeuralODEs Might be from Obfuscated Gradients". In: *arXiv preprint arXiv:2009.13145* (2020).

[9] Sergey Ioffe and Christian Szegedy. "Batch normalization: Accelerating deep network training by reducing internal covariate shift". In: *International conference on machine learning*. PMLR. 2015, pp. 448–456.

[10] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. "Adversarial machine learning at scale". In: *arXiv preprint arXiv:1611.01236* (2016).

[11] Xiang Li et al. "Understanding the disharmony between dropout and batch normalization by variance shift". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 2682–2690.

[12] Aleksander Madry et al. "Towards deep learning models resistant to adversarial attacks". In: *arXiv preprint arXiv:1706.06083* (2017).

[13] Yutian Pang et al. "Evaluating the Robustness of Bayesian Neural Networks Against Different Types of Attacks". In: *arXiv preprint arXiv:2106.09223* (2021).

[14] Nicolas Papernot et al. "The limitations of deep learning in adversarial settings". In: *2016 IEEE European symposium on security and privacy (EuroS&P)*. IEEE. 2016, pp. 372–387.

[15] Bernardo Ávila Pires and Csaba Szepesvári. "Multiclass classification calibration functions". In: *arXiv preprint arXiv:1609.06385* (2016).

[16] Dong Su et al. "Is Robustness the Cost of Accuracy?–A Comprehensive Study on the Robustness of 18 Deep Image Classification Models". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 631–648.

[17] Christian Szegedy et al. "Intriguing properties of neural networks". In: *arXiv preprint arXiv:1312.6199* (2013).

[18] Binyu Tian et al. "Bias field poses a threat to dnn-based x-ray recognition". In: *2021 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE. 2021, pp. 1–6.

[19] Dimitris Tsipras et al. "Robustness may be at odds with accuracy". In: *arXiv preprint arXiv:1805.12152* (2018).

[20] Xiaosen Wang et al. "At-gan: An adversarial generator model for non-constrained adversarial examples". In: *arXiv preprint arXiv:1904.07793* (2019).

[21] Yao-Yuan Yang et al. "A closer look at accuracy vs. robustness". In: *arXiv preprint arXiv:2003.02460* (2020).

[22] C Zhang et al. "Understanding deep learning requires rethinking generalization Proc. 5th Int". In: *Conf. on Learning Representations (ICLR)(Toulon, France, 28-26 April 2017) arXiv*. Vol. 1611. 2017.

[23] Hongyang Zhang et al. "Theoretically principled trade-off between robustness and accuracy". In: *International Conference on Machine Learning*. PMLR. 2019, pp. 7472–7482.